POMDPs and Information Gathering

Prof. Scott Niekum

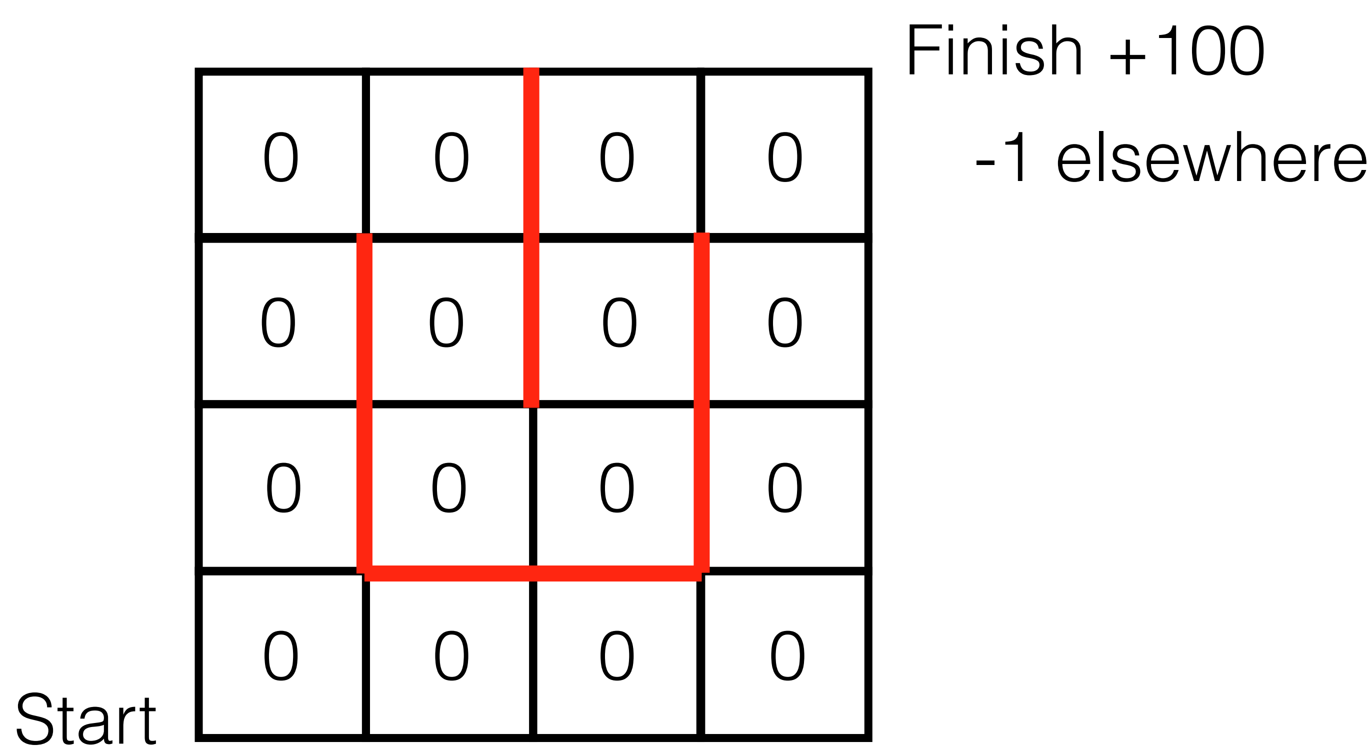
Markov Decision Process

Recall, MDP $M =< S, A, T, R >$

- If **T** unknown, model-free methods like Q-learning
- If knowledge of **T**, model-based dynamic programming

Bellman equation: $V_\pi(s) = \sum_a \pi(s,a) \sum_{s'} T(s,a,s') \left[R(s,a) + \gamma V_\pi(s')\right]$

Value iteration: $\hat{V}^*(s) = \max_a \sum_{s'} T(s,a,s') \left[R(s,a) + \gamma V_\pi(s')\right]$



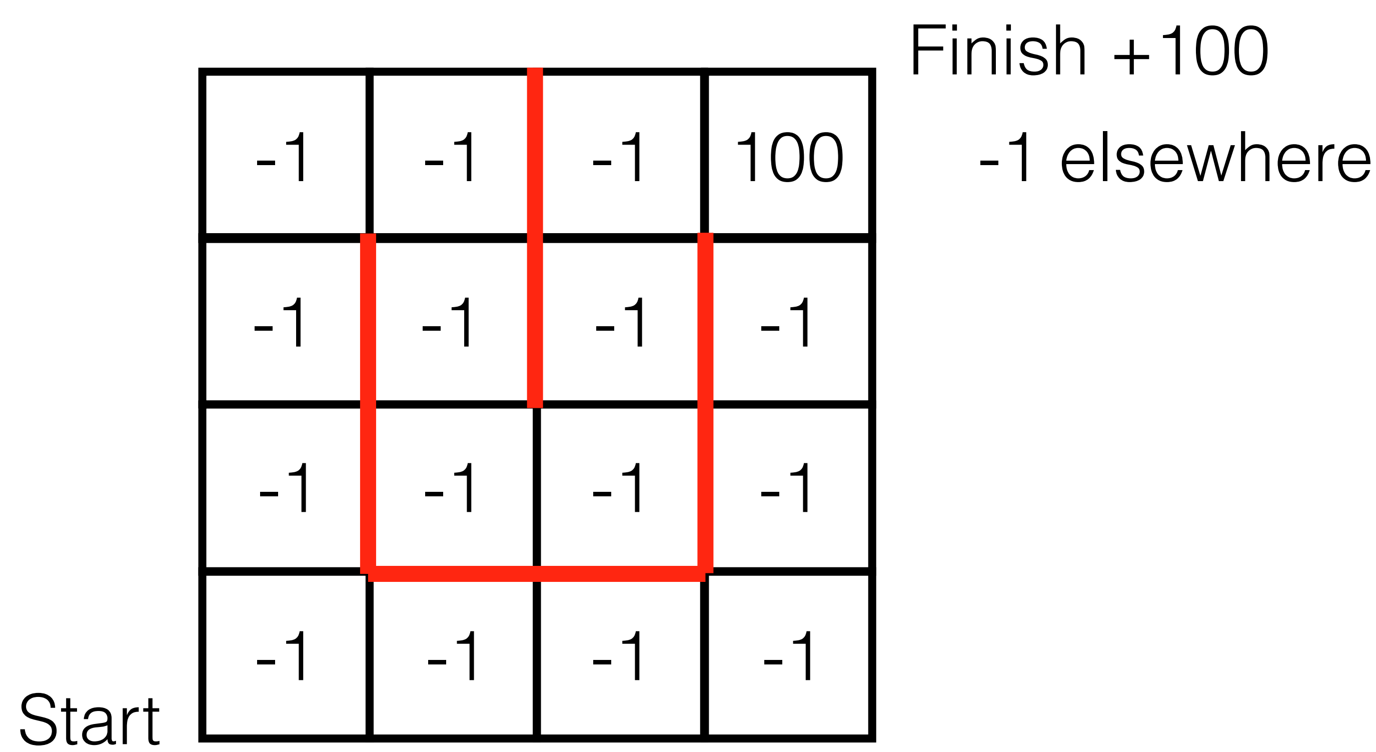
Markov Decision Process

Recall, MDP $M = < S, A, T, R >$

- If **T** unknown, model-free methods like Q-learning
- If knowledge of **T**, model-based dynamic programming

Bellman equation: $$V_\pi(s) = \sum_a \pi(s,a) \sum_{s'} T(s,a,s') \left[R(s,a) + \gamma V_\pi(s')\right]$$

Value iteration: $$\hat{V}^*(s) = \max_a \sum_{s'} T(s,a,s') \left[R(s,a) + \gamma V_\pi(s')\right]$$



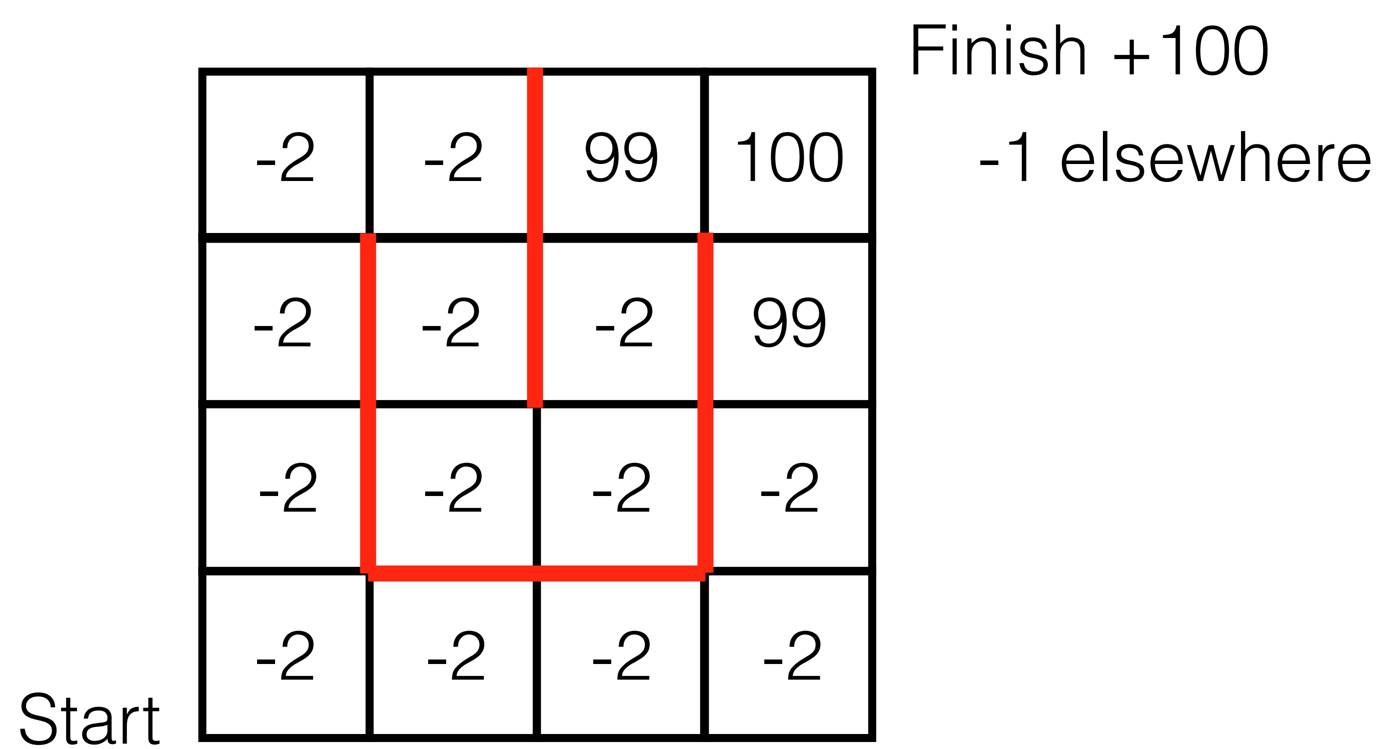
Markov Decision Process

Recall, MDP $M = < S, A, T, R >$

- If **T** unknown, model-free methods like Q-learning
- If knowledge of **T**, model-based dynamic programming

Bellman equation: $$V_\pi(s) = \sum_a \pi(s,a) \sum_{s'} T(s,a,s') \left[R(s,a) + \gamma V_\pi(s')\right]$$

Value iteration: $$\hat{V}^*(s) = \max_a \sum_{s'} T(s,a,s') \left[R(s,a) + \gamma V_\pi(s')\right]$$



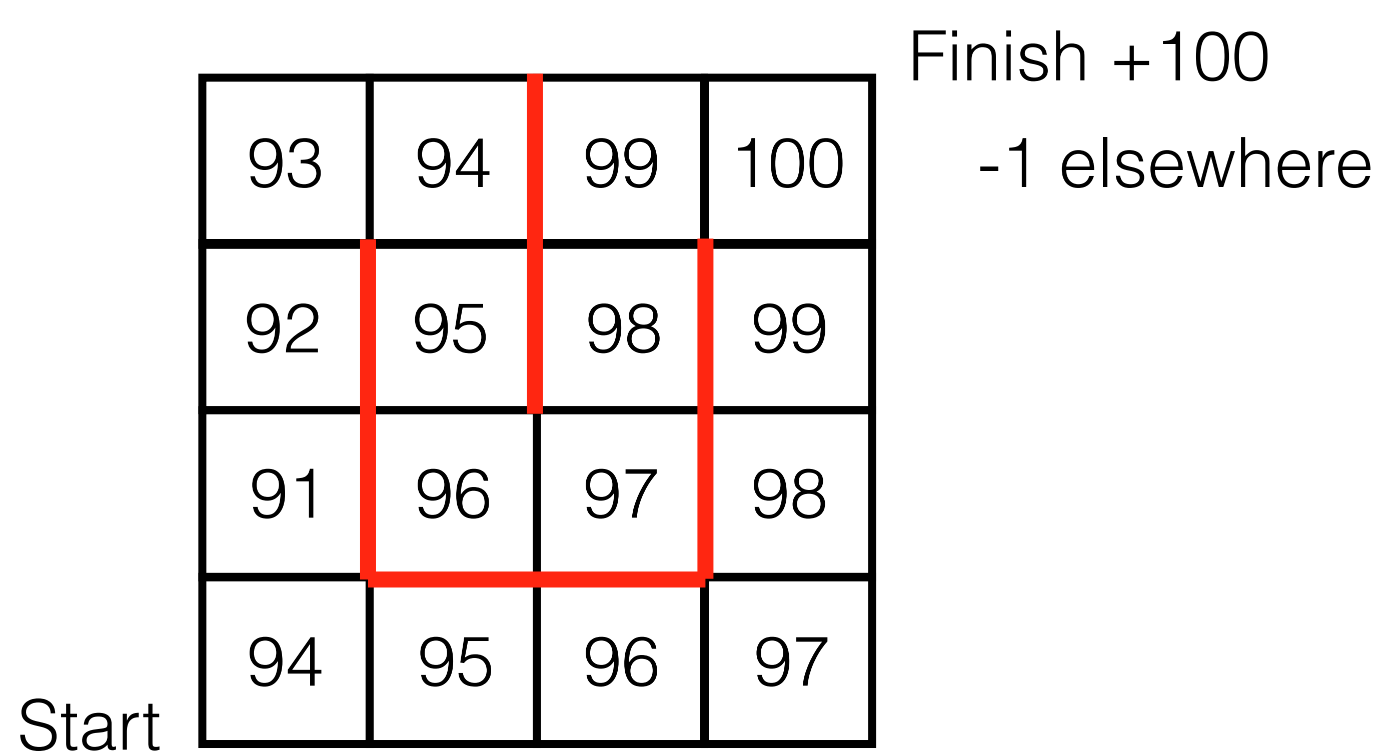
Markov Decision Process

Recall, MDP $M = < S, A, T, R >$

- If **T** unknown, model-free methods like Q-learning
- If knowledge of **T**, model-based dynamic programming

Bellman equation: $V_\pi(s) = \sum_a \pi(s,a) \sum_{s'} T(s,a,s') \left[R(s,a) + \gamma V_\pi(s') \right]$

Value iteration: $\hat{V}^*(s) = \max_a \sum_{s'} T(s,a,s') \left[R(s,a) + \gamma V_\pi(s') \right]$



Partially Observable Markov Decision Process

POMDP $P =< S, A, T, R, \Omega, O >$

Ω : a set of observations

$O : S \times \Omega \rightarrow [0, 1]$ observation probabilities

Examples when state is pose:

- GPS provides pose corrupted by noise
- Pose is inaccessible, only have odometer velocity

Instead of state, use belief state: $p(s|o_1, a_1, \ldots, o_N, a_N)$

Partially Observable Markov Decision Process

POMDP $P =< S, A, T, R, \Omega, O >$

Ω : a set of observations

$O : S \times \Omega \rightarrow [0, 1]$ observation probabilities

Examples when state is pose:

- GPS provides pose corrupted by noise
- Pose is inaccessible, only have odometer velocity

Instead of state, use belief state: $p(s|o_1, a_1, \ldots, o_N, a_N)$

Why can't we (tractably) use value iteration in a POMDP?

Partially Observable Markov Decision Process

Every probability discretized into 2 bins: 0-50% and 50-100%

$2^{|S|}$ possible belief states and can't handle continuous state!

Hack: Assume max likelihood state is the true state

- Doesn't take uncertainty into account.
- Lose nice properties of POMDPs like info gathering

Alternative: Only use parameterized belief states like Gaussian

- How to update mean and covariance?

Control problems

LQR - optimal control for linear dynamics + quadratic cost

$$x_{t+1} = Ax_t + Bu_t \qquad g(x_t, u_t) = x_t^\top Q x_t + u_t^\top R u_t$$

LQG - linear dynamics + quadratic cost + Gaussian noise

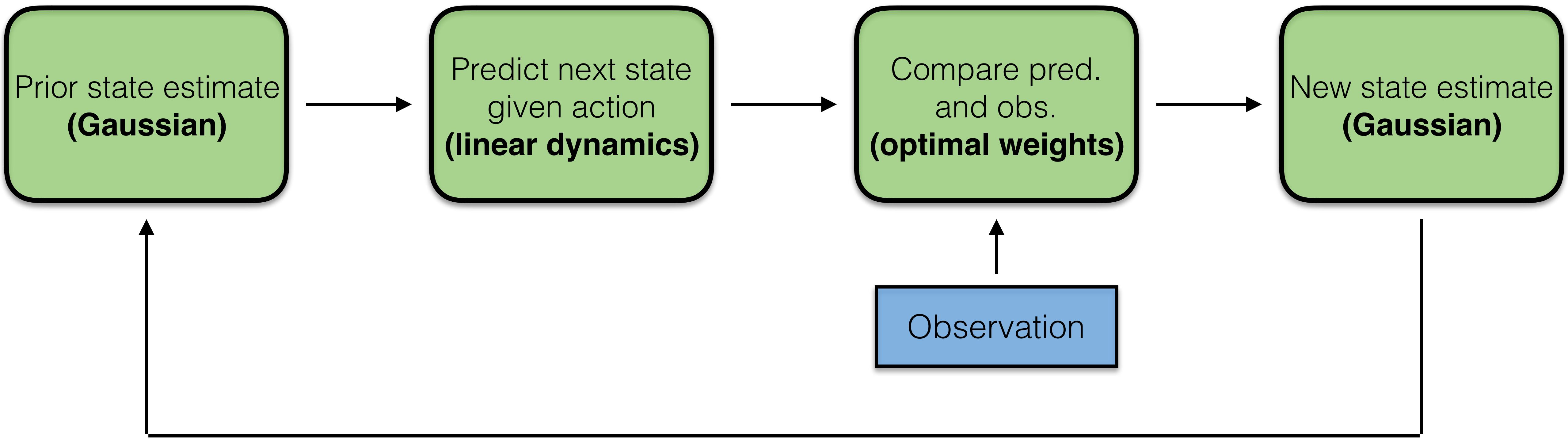
- LQR + Kalman filter point estimate

$$x_{t+1} = Ax_t + Bu_t + w_t$$
$$y_t = Cx_t + w_t$$

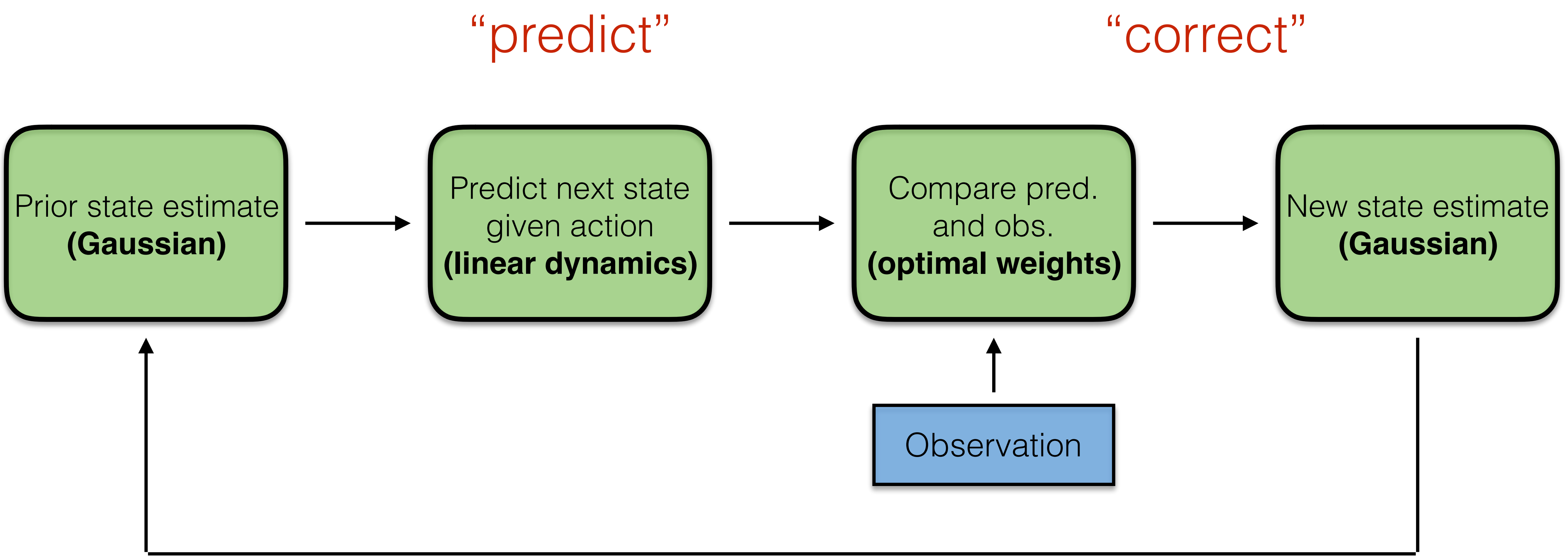
Platt et al. - B-LQR for planning in belief space in nonlinear POMDP

- LQR + Kalman filter full distribution
- Consider uncertainty directly → information gathering actions

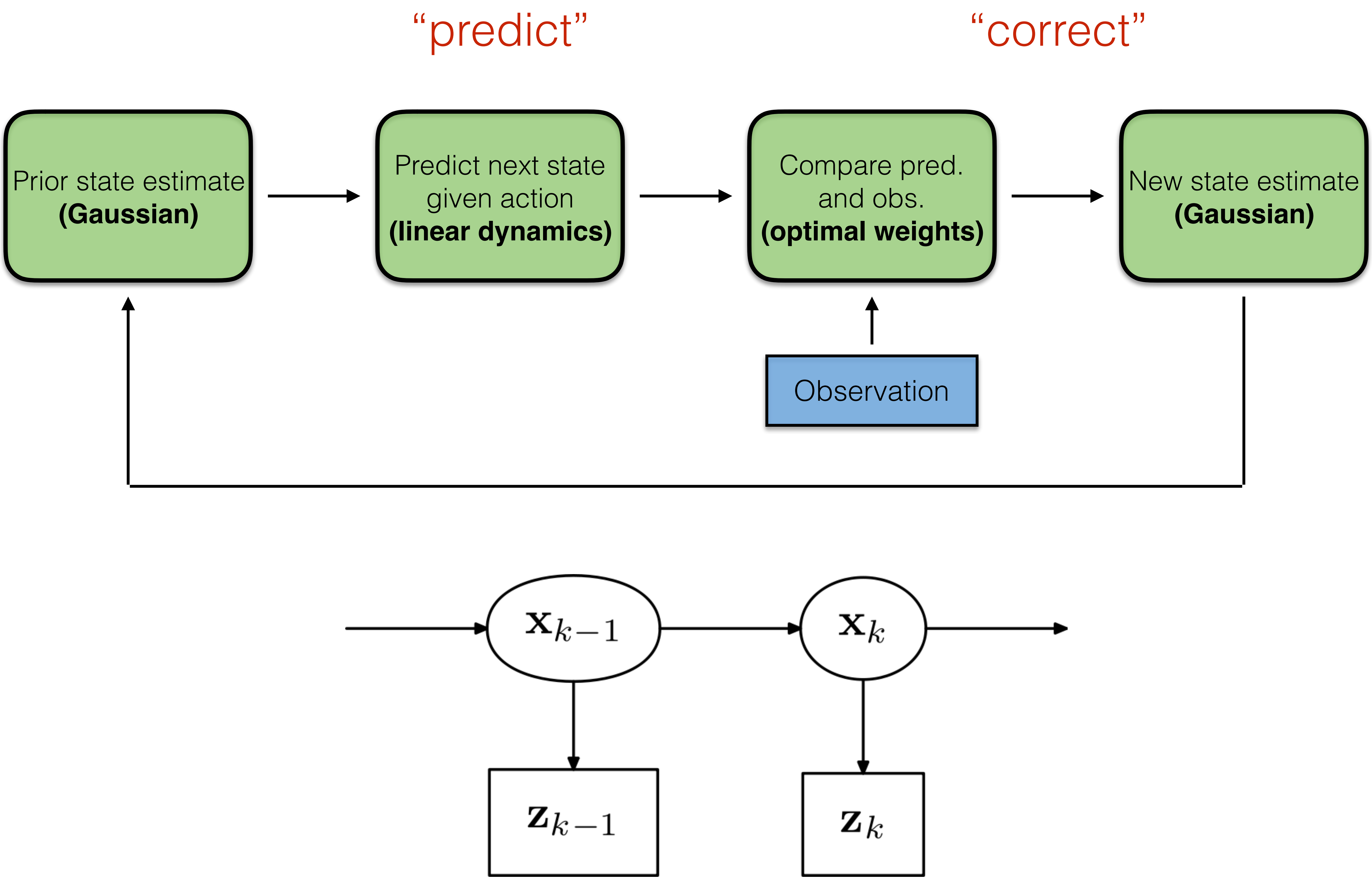
Kalman filter: conceptual idea



Kalman filter: conceptual idea



Kalman filter: conceptual idea



Extended Kalman filter

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad \longrightarrow \quad \boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}) + \boldsymbol{w}_{k-1}$$

$$z_k = Hx_k + v_k \quad \longrightarrow \quad \boldsymbol{z}_k = h(\boldsymbol{x}_k) + \boldsymbol{v}_k$$

$$\boldsymbol{F}_{k-1} = \left.\frac{\partial f}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}_{k-1}}$$

$$\boldsymbol{H}_k = \left.\frac{\partial h}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{k|k-1}}$$

Bringing it together

Extended Kalman filter - track state under nonlinear dynamics

LQG - Solve optimal control problem in **state space** with linear dynamics by using KF state estimate with LQR

Essentially a point estimate of state

B-LQR - Solve control problem in **nonlinear belief space**

Takes uncertainty directly into account for planning